

#2

Attorney Docket No. 1086.1158

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Masahisa Tamura, et al.

Application No.: TBA

Group Art Unit: TBA

Filed: March 4, 2002

Examiner: TBA

For: FILE SYSTEM, CONTROL METHOD, AND PROGRAM

11046 U.S. PTO
10/086894
03/04/02

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2001-350986

Filed: November 16, 2001

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing date(s) as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: March 4, 2002

By: John C. Garvey

John C. Garvey
Registration No. 28,607

700 11th Street, N.W., Ste. 500
Washington, D.C. 20001
(202) 434-1500

日 本 国 特 許 庁
JAPAN PATENT OFFICE



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日
Date of Application:

2001年11月16日

出 願 番 号
Application Number:

特願2001-350986

[ST.10/C]:

[JP2001-350986]

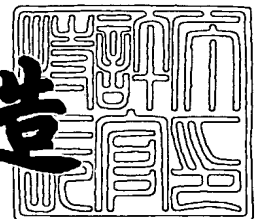
出 願 人
Applicant(s):

富士通株式会社

2002年 1月18日

特 許 庁 長 官
Commissioner,
Japan Patent Office

及 川 耕 造



出証番号 出証特2002-3000179

【書類名】 特許願

【整理番号】 0152036

【提出日】 平成13年11月16日

【あて先】 特許庁長官殿

【発明の名称】 ファイルシステム、制御方法及びプログラム

【請求項の数】 10

【国際特許分類】 G06F 12/00

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 田村 雅寿

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 武 理一郎

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 野口 泰生

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 荻原 一隆

【特許出願人】

 【識別番号】 000005223

 【氏名又は名称】 富士通株式会社

【代理人】

 【識別番号】 100079359

 【弁理士】

【氏名又は名称】 竹内 進

【手数料の表示】

【予納台帳番号】 009287

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9704823

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ファイルシステム、制御方法及びプログラム

【特許請求の範囲】

【請求項1】

オペレーティングシステム上で動作するファイルシステムに於いて、
ファイルに対するアクセスが発生した場合に、該アクセスに応じて前記ファイルを処理するアクセス実行部と、
ユーザが予め定義したユーザ定義処理を保持する定義処理保持部と、
前記ファイルへのアクセスを契機として、前記ユーザ定義処理を実行させる定義処理実行部と、
を備えたことを特徴とするファイルシステム。

【請求項2】

請求項1記載のファイルシステムに於いて、
前記ファイルをデータ領域とメタデータ領域との2つの領域で管理するファイル管理部を有し、更に前記メタデータ領域には拡張メタデータ領域が設けられ、
前記ファイル管理部は、ユーザにより前記拡張メタデータ領域の形式を指定可能とすることを特徴とするファイルシステム。

【請求項3】

請求項2記載のファイルシステムに於いて、前記ファイル管理部は、前記データ領域の内容に応じて前記拡張メタデータ領域の形式を指定することを特徴とするファイルシステム。

【請求項4】

請求項3記載のファイルシステムに於いて、前記ファイル管理部は、前記拡張メタデータ領域の形式としてファイルタイプというメタデータを設定し、前記

ファイルタイプに応じて前記拡張メタデータ領域の形式を決定することを特徴とするファイルシステム。

【請求項 5】

請求項 4 記載のファイルシステムに於いて、前記ファイル管理部は、ファイルの作成時にファイルタイプを決定すると共に該ファイルタイプに応じて拡張メタデータ領域を設定し、その後、ユーザによる前記ファイルタイプの変更と前記拡張メタデータ領域の変更を可能とすることを特徴とするファイルシステム。

【請求項 6】

請求項 2 記載のファイルシステムに於いて、前記定義処理実行部は、前記ファイルのデータ領域への書き込みの際に、前記ユーザ定義処理に基づき前記メタデータ領域の拡張メタデータを変更することを特徴とするファイルシステム。

【請求項 7】

請求項 2 記載のファイルシステムに於いて、前記定義処理実行部は、前記ファイルのデータ領域への書き込みの際に、別途設けられたユーザプログラムにメッセージを送付して前記拡張メタデータ領域の拡張メタデータを変更させることを特徴とするファイルシステム。

【請求項 8】

請求項 2 記載のファイルシステムに於いて、前記定義処理実行部は、前記ファイルのデータ領域への書き込みの際にデータ領域変更フラグを立て、別途設けられたユーザプログラムにより前記フラグが立ったことを契機として前記拡張メタデータ領域の拡張メタデータを変更させることを特徴とするファイルシステム。

【請求項 9】

オペレーティングシステム上で動作するファイルシステムの制御方法に於いて

ユーザが予め定義したユーザ定義処理を保持する定義処理保持ステップと、
ファイルに対するアクセスが発生した場合に、該アクセスに応じて前記ファイルを処理するアクセス実行ステップと
前記ファイルへのアクセスを契機として、前記ユーザ定義処理を実行させる定義処理実行ステップと、
を有することを特徴とするファイルシステムの制御方法。

【請求項10】

コンピュータに、
ユーザが予め定義したユーザ定義処理を保持する定義処理保持ステップと、
ファイルに対するアクセスが発生した場合に、該アクセスに応じて前記ファイルを処理するアクセス実行ステップと
前記ファイルへのアクセスを契機として、前記ユーザ定義処理を実行させる定義処理実行ステップと、
を実行させることを特徴とするプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、オペレーティングシステム上で動作するファイルシステム、制御方法及びプログラムに関し、特にファイルへのアクセスを契機としてユーザが定義した処理を実行可能とするファイルシステム、制御方法及びプログラムに関する。

【0002】

【従来の技術】

従来のファイルシステムでは、ファイルシステムの設計者がその動作やメタデータを定義し、ファイルシステムは設計者が定義した動作を行っている。そして

、同じファイルシステム上のファイルは、ファイルシステムで決められた形式のメタデータを持ち、ファイルシステムで決められた処理を行っている。

【0003】

【発明が解決しようとする課題】

このような従来のファイルシステムの上で、ファイルへのアクセスを契機にユーザが定義した処理を実行させようとする、特定のアプリケーション内にて何らかの処理を契機としてユーザが定義した処理を実行できるように組む必要が出てくる。

【0004】

また、ファイルにユーザが定義したメタデータを保持させようとする、アプリケーションが特定のファイルに対し別のファイルを用意し、その別ファイルを元ファイルユーザが定義したメタデータをデータ領域に保持するメタデータファイルとして使用する必要がある。

【0005】

こういった従来の手法では、異なるアプリケーションからのアクセスに対して一貫性が保障されず、また柔軟性に欠けることとなる。そのうえ、ファイルシステムとアプリケーションの間を行き来する必要が生じるため、性能上の問題も生じることとなる。

【0006】

本発明は、アプリケーションに依存することなくファイルへのアクセスを契機にユーザが定義した処理を可能とすることで、より柔軟なデータ制御を可能とし、また性能の向上を可能とするファイルシステム、制御方法及びプログラムを提供することを目的とする。

【0007】

【課題を解決するための手段】

図1は本発明の原理説明図である。本発明は、オペレーティングシステム上で動作するファイルシステムに於いて、ファイルに対する書込み又は読出しといっ

たアクセスが発生した場合に、アクセスに応じてファイルを処理するアクセス実行部20と、ユーザが予め定義したユーザ定義処理を保持するユーザ定義処理保持部（定義処理保持部）15と、ファイルへのアクセスを契機としてユーザ定義処理を実行させる定義処理実行部22とを備えたことを特徴とする。

【0008】

このように本発明は、ファイルに対するアクセスを契機にユーザが定義した処理をファイルシステムが直接実行することとなり、これによりファイル内容やアクセスに応じてユーザが定義したメタデータの保持やユーザが定義した処理を実行できる。

【0009】

ここで、ファイルシステム12は、図1（B）のように、ファイル14をデータ領域30とメタデータ領域32との2つの領域で管理するファイル管理部18を有し、更にメタデータ領域32には拡張メタデータ領域34が設けられ、ファイル管理部18は、ユーザにより拡張メタデータ領域34の形式を指定可能とする。

【0010】

ファイル管理部18は、データ領域30の内容に応じて拡張メタデータ領域34の形式を指定する。具体的にはファイル管理部18は、拡張メタデータ領域34の形式としてファイルタイプというメタデータを設定し、このファイルタイプに応じて拡張メタデータ領域34の形式を決定する。

【0011】

ファイル管理部18は、ファイルの作成時にファイルタイプを決定すると共にこのファイルタイプに応じた拡張メタデータ領域34を設定し、その後、ユーザによるファイルタイプの変更と前記拡張メタデータ領域34の変更を可能とする。ファイル管理部18は、データ領域30の内容からファイルタイプを自動的に決定すると共に拡張メタデータ領域34も自動的に決定する。

【0012】

ファイル管理部18は、複数ファイルを管理するためのディレクトリを用いたツリー構造を備え、ファイル作成時に設定されるファイルタイプは、親ディレク

トリのファイルタイプを継承する。

【0013】

ファイル管理部18は、拡張メタデータ領域34に、データ領域30から抽出した情報を拡張メタデータとして保持する。データ領域30から抽出された拡張メタデータは、データ領域30と拡張メタデータ領域34とに二重化して保持する。

【0014】

データ領域30から抽出した拡張メタデータは、実体は前記データ領域に持ち、拡張メタデータ領域には前記データ領域の実体へのポインタを保持するようにしても良い。

【0015】

定義処理実行部22は、ファイル14のデータ領域30への書き込みの際に、ユーザ定義処理に基づき拡張メタデータ領域34の拡張メタデータを変更する。

【0016】

定義処理実行部22は、ファイル14のデータ領域30への書き込みの際に、別途設けられたユーザプログラムにメッセージを送付して拡張メタデータ領域34の拡張メタデータを変更させる。

【0017】

定義処理実行部22は、ファイル14のデータ領域30への書き込みの際にデータ領域変更フラグを立て、別途設けられたユーザプログラムによりフラグが立ったことを契機として拡張メタデータ領域34の拡張メタデータを変更させる。

【0018】

定義処理実行部22は、ファイルタイプに応じてユーザ定義処理を実行する。

【0019】

ファイルシステムは、ユーザが処理を定義するためのAPIを備える。このAPIは、カーネル領域で実行されるAPIとユーザ領域で実行されるAPIとの2層構造を備える。拡張メタデータ領域34のサイズはファイル内容に応じて可変である。

【0020】

本発明は、オペレーティングシステム上で動作するファイルシステムの制御方法を提供する。この制御方法は、

ユーザが予め定義したユーザ定義処理を保持する定義処理保持ステップと、

ファイルに対するアクセスが発生した場合に、このアクセスに応じてファイル
を処理するアクセス実行ステップと

ファイルへのアクセスを契機として、ユーザ定義処理を実行させる定義処理実行
ステップと、

を有することを特徴とする。

【 0 0 2 1 】

本発明は、オペレーティングシステム上で動作するファイルシステムを実現する
プログラムを提供する。このプログラムは、

コンピュータに、

ユーザが予め定義したユーザ定義処理を保持する定義処理保持ステップと、

ファイルに対するアクセスが発生した場合に、このアクセスに応じてファイル
を処理するアクセス実行ステップと

ファイルへのアクセスを契機として、ユーザ定義処理を実行させる定義処理実行
ステップと、

を実行させることを特徴とする。

【 0 0 2 2 】

このファイルシステムの制御方法及びプログラムの詳細は、ファイルシステム
の場合と同様になる。

【 0 0 2 3 】

【発明の実施の形態】

図 2 は本発明によるファイルシステムの機能構成のブロック図である。図 2 に
おいて、コンピュータのオペレーティングシステム 1 0 は永続的な記憶媒体の利用
方法を提供し、データを管理する必要がある。このため、UNIX (R) や W
i n d o w s (R) などのオペレーティングシステムにあっては、データをファ
イルという単位で抽象化してデータ管理を行い、ユーザはオペレーティングシス

テム10上で動作するファイルシステム12を用いてデータをアクセスする。

【0024】

本発明のファイルシステム12は、ユーザが作成したファイル14を処理するため、ファイル管理部18、アクセス実行部20及び定義処理実行部22を備えている。このような本発明のファイルシステム12にあっては、ファイル14に対するユーザ24-1、24-2側からのアクセスを契機として、ファイル14に対しユーザが予め定義したユーザ定義処理を実行させることを可能とする。このユーザからのアクセスを契機に実行されるユーザ定義処理は、ユーザ定義処理保持部15に、ファイル14に関連付けられて保持されている。

【0025】

アクセス実行部20は、ユーザ24-1、24-2によるファイル14に対するアクセス、具体的には書込みまたは読出しが発生した場合に、このアクセスに応じてファイル14に対する書込みまたは読出しの処理を実行する。定義処理実行部22は、アクセス実行部20の処理を引き起こした同じアクセスを契機に、ファイル14に関連付けられているユーザ定義処理保持部15に保持している対応するユーザ定義処理を実行させる。

【0026】

図3は、図2のファイルシステムとハードウェアの関係である。コンピュータにおける記憶装置としては、メモリ、ハードディスクドライブなど様々なハードウェア28-1～28-nが用いられている。

【0027】

ファイルシステム12は、ハードウェア28-1～28-nのそれぞれに用意されたソフトウェアであるドライバ26-1～26-nに対し共通API25を介して接続され、ドライバ26-1～26-nのそれぞれを用いて、対応するハードウェア28-1～28-nにアクセスする。これによりファイルシステム12は、個々のハードウェア28-1～28-nを意識することなく実装することができる。

【0028】

図4は、本発明のファイルシステム12におけるユーザからのアクセスに対す

る基本的な処理を表わしている。本発明のファイルシステムにあっては、ファイル14-1、14-2はアクセスに応じてユーザが定義した複数のユーザ定義処理16-1～16-4と関連付けられている。

【0029】

この場合、ファイル14-1はアクセスXを受けた際にアクセス種別が「A」または「B」かを判別し、アクセスX=アクセスAであればユーザ定義処理16-3に関連付けられており、またアクセスX=アクセスBであればユーザ定義処理16-1に関連付けられている。

【0030】

またファイル14-2についても、アクセスYに対しアクセス種別「A」または「B」を判別し、アクセスY=アクセスAであればユーザ定義処理16-3に関連付けられており、またアクセスY=アクセスBであればユーザ定義処理16-4に関連付けられている。

【0031】

このため、ファイル14-1に対し「アクセスA」というアクセスが発行された場合、本発明のファイルシステムはファイル14-1の「アクセスA」に対応する処理を実行すると共に、ユーザが「アクセスA」のために予め定義した処理であるユーザ定義処理16-3を実行する。

【0032】

図5は、本発明のファイルシステムによりユーザ定義処理を実行する場合と、従来のアプリケーションを使用する場合を対比して示している。

【0033】

図5(A)は従来システムである。従来システムは、ユーザ206-1とオペレーティングシステム200上で動作するファイルシステム202のファイル204の間にアプリケーション208を配置し、このアプリケーション208がファイル204のアクセスを監視することで、ユーザが定義した処理として例えば外部に通知する処理を実行する機能を持たせている。

【0034】

しかしながら、アプリケーション208を経由してアクセスするユーザ206

ー 1 に対しては希望どおりのユーザが定義した処理を実行することができるが、アプリケーション 2 0 8 を経由せずにアクセスしてくるユーザ 2 0 6 - 2 に対してはユーザが定義した処理を実行することができない。また、アプリケーション 2 0 8 がアクセスに介在することによってオーバーヘッドが生ずる問題がある。

【 0 0 3 5 】

これに対し図 5 (B) の本発明にあっては、ユーザ定義処理 1 6 をファイルシステム 1 2 側に実装することにより、ユーザ 2 4 - 1 , 2 4 - 2 のアクセスに対し特定のアプリケーションに依存せず、ユーザ定義処理 1 6 の実行によりユーザが予め処理として設定した外部に対する通知を可能とし、アプリケーション経由によるオーバーヘッド分を削除することができる。

【 0 0 3 6 】

図 6 は、本発明のファイルシステムにおける処理手順、即ちプログラムのフローチャートである。図 6 において、まずステップ S 1 でユーザにより定義された処理をユーザ定義処理保持部 1 5 に保存する。続いてステップ S 2 でファイルアクセスの有無をチェックしており、ファイルアクセスがあると、ステップ S 3 でアクセスに応じたファイル処理、即ち書込みまたは読出しを実行する。

【 0 0 3 7 】

続いてステップ S 4 でアクセス種別に応じて関連付けられたユーザ定義処理を選択し、ステップ S 5 で選択したユーザの定義した処理を実行する。このようなステップ S 2 ~ S 5 の処理を、ステップ S 6 でシステムの停止指示があるまで繰り返す。

【 0 0 3 8 】

図 7 は、本発明のファイルシステムの具体的な実施形態であり、この実施形態にあっては拡張メタデータをもつファイルシステムを例にとっている。

【 0 0 3 9 】

図 7 において、ファイル 1 4 はデータ領域 3 0 とメタデータ領域 3 2 の 2 つの領域で管理されている。データ領域 3 0 にはファイル 1 4 のデータの実態が記憶される。メタデータ領域 3 2 にはファイル 1 4 の管理情報、例えば更新時刻やファイルサイズなどが記憶されている。

【0040】

本発明にあっては、メタデータ領域32にユーザが定義可能である拡張メタデータ領域34を新たに設けている。従来よりメタデータ領域32はファイルシステムごとに形式が決まっており、ファイルシステムの設計者が作成していたが、本発明で新たに設けられた拡張メタデータ領域34はユーザが定義することが可能であり、またファイル14の内容によって、異なる形式の拡張メタデータを保持することを可能とする。

【0041】

例えば拡張メタデータ領域34には拡張メタデータ36-1, 36-2, 36-3が設定されており、これらの拡張メタデータ36-1~36-3に関連付けられて、ユーザ定義処理保持部15にユーザが予め定義した処理であるユーザ定義処理16-1~16-3が格納されている。

【0042】

図8は、ファイル内容に応じて拡張メタデータ領域の形式を変化させるための実装方法の一例として、ファイルタイプを実装した場合の実施形態である。

【0043】

図8において、ファイル14は、メタデータ領域32に設けた拡張メタデータ領域34の指定された位置にファイルタイプ38というメタデータを保持する。そして拡張メタデータ領域34のファイルタイプ38に応じて異なった形式の拡張メタデータをユーザが定義し、ユーザが定義した形式の拡張メタデータを拡張メタデータ領域34に保持することになる。

【0044】

更にユーザは、ファイルタイプ38に応じてユーザ定義処理を関連付けていく。例えばファイル14には、ファイルタイプBが設定されていたとする。このファイルタイプBに定義された拡張メタデータを保持し、ユーザ定義処理保持部15のユーザ定義処理16-1, 16-2, ...を関連付ける。

【0045】

拡張メタデータ領域34のファイルタイプ38はユーザが定義することが可能であり、また動的にファイルタイプを加えることを可能とする。この動的にファ

イルタイプを加えるための実装方法の1つとして、ファイルタイプの種類をリスト構造で管理し、動的な追加と削除を可能とする。

【0046】

図9は、リスト構造で管理されるファイルタイプ構造体42の例である。ファイルタイプ構造体42には、リスト構造のためのポインタ「next」とファイルタイプの名前「name」及び、全てのファイルタイプに対し一意であるナンバー「magic」によって構成され、リスト構造にて管理される。

【0047】

図10は、図9のファイルタイプ構造体42をリスト構造で管理するファイルタイプリストの処理の流れである。図10(A)は最初のファイル作成時であり、ファイルタイプ構造体44-1が作られる。

【0048】

図10(B)はファイルタイプMailを登録した場合であり、リスト構造でファイルタイプMailのファイルタイプ構造体44-2がリンクされる。図10(C)は更にファイルタイプNewsを登録した場合であり、対応するファイルタイプ構造体44-3がリンクされる。

【0049】

更に図10(D)はファイルMailを削除した場合である。この図10の処理のため、ファイルタイプ登録及び削除用の関数を準備し、この関数の実行により図10のようなリストの動的な追加と削除を行うことができる。

【0050】

また本発明のファイルシステムで管理するファイルのファイルタイプは、ファイル作成時にデフォルトのファイルタイプを設定しておき、その後に必要に応じて変更するという手順をとることを可能とする。この場合、一般的なファイルタイプを予め用意しておき、ファイル作成時には一般的なファイルタイプをデフォルトのファイルタイプとして設定することもできる。

【0051】

またファイルの管理に当たり、ディレクトリを用いたツリー構造を用いて、デフォルトのファイルタイプは、そのファイルの親ディレクトリのファイルタイ

プを承継する方法をとることもできる。更に、ファイルの内容を読み取り、自動的に最適なファイルタイプを設定するようにしてもよい。

【0052】

即ち、ファイルの作成手順としては

(1) ファイル作成時にデフォルトのファイルタイプを設定し、同時にデフォルトのファイルタイプを拡張メタデータにも設定する、

(2) 必要に応じてファイルタイプを設定する、

といった二段構えで処理される。このためファイル作成時には、ユーザが意識することなくファイルタイプが設定されることになる。

【0053】

拡張メタデータ領域34の拡張メタデータの中にはファイルのデータ領域の内容を抽出した情報を保持することもできる。この場合の拡張メタデータの保持の方法として、図11のように、拡張メタデータ領域34の内部にデータ領域30の内容となる抽出情報48を拡張メタデータ50としてコピーし、データを二重化して保持する。この二重化により保持された場合にあっては、データ領域30が変更された際には拡張メタデータ領域34への変更後の抽出情報48のコピーを必要とする。

【0054】

図12は、拡張メタデータの中にファイルのデータ領域の内容を抽出した情報を保持する他の実施形態であり、この実施形態にあっては、拡張メタデータ領域34にはデータ領域30の実態である抽出情報48を保持せず、抽出情報48へのポインタ52を拡張メタデータとして保持する。この場合、データ領域30が変更された際には拡張メタデータ領域34のポインタ52及びそのサイズ情報の更新が必要となる。

【0055】

このような図11、図12のファイルのデータ領域の内容を拡張メタデータとして保持する具体例としては、例えばファイルタイプとして「Mail」の形式を考えると、データ領域の内のメールヘッダ部分の例えば「To」といった領域を拡張メタデータ領域34に保持することが考えられる。メールヘッダ部分の「

「To」領域は、差出人がメールの送信時に宛先に指定した名前やメールアドレスが格納されている。

【0056】

図11または図12の実施形態に基づきデータ領域30の内容を拡張メタデータ領域34に反映している場合、データ領域30の変更に対し拡張メタデータ領域34の変更をする必要があり、その際に同期の問題が出てくる。

【0057】

図13は、データ領域の変更に対し拡張メタデータ領域を変更する場合の処理の実施形態である。図13にあっては、ファイル14のデータ領域30に対する書込み56の際に、ユーザ定義処理16に基づいて拡張メタデータ54を変更することを特徴とする。

【0058】

即ち、ファイル14のデータ領域30への書込み56が起きると、図2に示した本発明のファイルシステム12のアクセス実行部20がデータ領域30の書込み処理を実行すると共に、図2の定義処理実行部22が拡張メタデータ54に関連付け58が行われているユーザ定義処理16を実行する。

【0059】

ユーザ定義処理16の中には、定義内容60として「拡張メタデータの変更」が定義されている。このため拡張メタデータ領域34の拡張メタデータ54は、データ領域30の変更に伴い、例えば図11の場合には変更後の抽出情報48がコピーされ、図12の場合には変更後の抽出情報48のポインタ52とそのサイズが変更される。

【0060】

なお、このデータ領域への書込み56に伴うデータ領域30の書込みとユーザ定義処理16の実行は連続して行われ、この一連の処理の中で拡張メタデータの変更が自動的に行われるため、拡張メタデータの同期に関しアプリケーションが意識する必要はない。但し、一連の書込みに伴う連続処理の間、ファイル14に対する他の処理はブロックされる。

【0061】

図 1 4 は、データ領域の書込みの際にファイルシステムはユーザ定義処理を実行して、別途設けられたユーザプログラムにメッセージを送り、メッセージを受け取ったユーザプログラムによって拡張メタデータの変更を行う実施形態である。

【 0 0 6 2 】

図 1 4 において、ファイル 1 4 のデータ領域 3 0 に書込み 6 4 が起きると、図 2 のファイルシステム 1 2 のアクセス実行部 2 0 がデータ領域 3 0 に対する書込みを実行し、続いて書込みアクセスに関連付けられたユーザ定義処理 1 6 を図 2 の定義処理実行部 2 2 が実行する。

【 0 0 6 3 】

このユーザ定義処理 1 6 の定義内容 6 8 にはメッセージ送信が定義されており、このため外部のユーザプログラム 7 0 に対しメッセージが送られる。メッセージを受け取ったユーザプログラム 7 0 は、プログラム内容 7 2 に「拡張メタデータの変更」が記述されていることから、ファイル 1 4 の拡張メタデータ領域 3 4 の拡張メタデータ 6 2 を、変更されたデータ領域 3 0 の内容に変更する。

【 0 0 6 4 】

このようにデータ領域の変更に伴い拡張メタデータを外部のユーザプログラムにより変更する処理は、アプリケーションレベルで処理ができるため、より柔軟な処理を行うことができるが、同期の一貫性についてはアプリケーション側で管理する必要がある。

【 0 0 6 5 】

図 1 5 は、データ領域の書込みの際にデータ領域変更フラグを立て、別途設けられたユーザプログラムがデータ領域変更プログラムを監視し、フラグが立ったことを契機にユーザプログラムが拡張メタデータを変更する処理を実行する実施形態である。

【 0 0 6 6 】

図 1 5 において、ファイル 1 4 のデータ領域 3 0 に書込みが起こると、図 2 のファイルシステム 1 2 のアクセス実行部 2 0 がデータ領域 3 0 に対するデータの書込みを実行し、次に図 2 の定義処理実行部 2 2 が書込み 7 8 のアクセスに関連

付け 80 が行われたユーザ定義処理 16 を実行する。

【0067】

このユーザ定義処理 16 の定義内容 82 には「フラグ設定」が定義されており、したがってユーザ定義処理 16 の実行で拡張メタデータ領域 34 に設けているデータ領域変更フラグ 76 が立てられる。

【0068】

ユーザプログラム 86 は拡張メタデータ領域 34 のデータ領域変更フラグ 76 の監視 90 を行っており、フラグが立っていることを認識した時点で、プログラム内容 88 として記述されている「拡張メタデータの変更」の処理を実行し、書込み後のデータ領域 30 のデータ内容を拡張メタデータ 74 にコピーするか、あるいはポインタの更新を行う。

【0069】

このように、ユーザ定義処理により制御されるフラグをユーザプログラムで監視してデータ書込みに伴う拡張メタデータの変更処理を行うことで、ファイルシステム側で行われる書込みから連続して行われる処理を最小限に抑えることができる。

【0070】

なお、データ領域変更フラグ 76 によって拡張メタデータ領域 34 の変更処理を排他的に行うことが可能であるが、ユーザプログラム 86 によりフラグを常に監視しなくてはならないことから、この分のオーバーヘッドは生ずることになる。

【0071】

以上説明した本発明のファイルシステムによるデータ領域の変更に対する処理は、拡張メタデータ領域の更新以外に、アクセスを契機として特定の処理を実行させる場合にも同様の手法を取ることができる。

【0072】

例えばデータ領域への書込みを契機として、予め定めておいて特定のユーザに対しファイルを転送するといった処理が実現できる。また書き込みアクセスだけではなく、読出しなどの他のアクセスに対しても同様の手法を適用できる。例えば、拡張メタデータ領域内に読出し回数を記録するカウンタを設けておき、この

カウンタが一定の数値に達したら、予め定めた特定のユーザに通知する処理を実行することもできる。

【 0 0 7 3 】

図 1 6 は本発明のファイルシステムの具体的な応用例であり、メールスプールファイルでの利用例として、メールサーバが受け取ったメールスプールファイルに追加書込みするメールから情報を抽出し、ユーザが設定した相手先に通知するシステムの実施形態である。

【 0 0 7 4 】

図 1 6 において、まずメールサーバ 9 2 に対しルール設定 9 8 を予め行っておく。このルール設定 9 8 は

- (1) a a a @ b b b . c c c . d d から来たメールは削除する。
- (2) a b c @ b b b . c c c . d d から来たメールであれば、x x . x x x . x x . x x という I P アドレスにメッセージを送る。
- (3) x y z @ b b b . c c c . d d から来たメールであれば、x x x @ a b . c d . e f にメールを転送する。
- (4) 本文に“返信希望”と書かれたメールが来た場合、読んだ際に自動的に差出人に確認メールを送る。

といったルールを設定しておく。

【 0 0 7 5 】

メールサーバ 9 2 は、新たにメールを受信するメールサーバ受信 1 0 0 が起きると、メールスプールファイル 9 4 に対し受信メールの追加書込み 9 6 を行う。この追加書込み 9 6 を行ったメールについて、ルール設定 9 8 の条件に合うか否か判別する。

【 0 0 7 6 】

この場合、例えば「a a a @ b b b . c c c . d d」からのメールであれば削除し、「a b c @ b b b . c c c . d d」からのメールであれば「x x . x x x . x x . x x」に対し通知し、また「x y z @ b b b . c c c . d d」から来たメールであれば「x x x @ a b . c d . e f」にメールを転送し、更に本文に“返信希望”という文字列があれば返信希望フラグを立てておく。

【0077】

これが設定した条件に合わせた動作102である。続いてユーザがクライアント104によりユーザメール読み込み106を行ったとすると、このときメールに返信希望フラグが立っていれば、差出人に対し受信確認のメールを送信する設定条件に合わせた動作108を実行することになる。

【0078】

更に、本発明のファイルシステムで設定したファイルタイプに基づくメタデータ情報を有効に利用するため、ユーザが定義する口となるAPIをファイルシステムを持つ必要がある。ユーザはファイルシステムの持つAPIを利用して、ユーザの定義処理や拡張メタデータの設定を行う。

【0079】

このためのAPIには図17のように、ユーザプログラム114に対し、カーネルモードのAPI110-1~110-nを備えたカーネルモードAPI110と、ユーザAPI112-1~112-mを備えたユーザモードAPI112の2層構造とすることが望ましい。

【0080】

このAPIには、ファイルタイプの追加・削除機能、ファイルタイプの設定・変更機能、拡張メタデータ形式の設定・変更機能、ユーザ定義処理の設定・変更機能、拡張メタデータ領域の書込み・読出し機能、データ領域と拡張メタデータ領域との同期機能などの機能が実装されることになる。

【0081】

更に本発明のファイルシステムにあっては、ユーザが定義可能な拡張メタデータ領域は、ユーザの定義によって、そのサイズが異なる。またファイルのデータ領域に応じて拡張メタデータ領域が増加し、不足することが考えられる。

【0082】

そこで図18(A)のように、当初のファイル14について設定された拡張メタデータ領域34が拡張メタデータの増加により、そこに収まらない場合には、図18(B)のように、新たに拡張メタデータ領域340を確保し、その間をリンクすることによって拡張メタデータ領域34の拡大を実現する。

【 0 0 8 3 】

なお、本発明は上記の実施形態に限定されず、その目的と利点を損なわない適宜の変形を含む。更に本発明は上記の実施形態に示した数値による限定は受けない。

【 0 0 8 4 】

(付記)

(付記 1)

オペレーティングシステム上で動作するファイルシステムに於いて、
ファイルに対するアクセスが発生した場合に、該アクセスに応じて前記ファイル
を処理するアクセス実行部と、
ユーザが予め定義したユーザ定義処理を保持する定義処理保持部と、
前記ファイルへのアクセスを契機として、前記ユーザ定義処理を実行させる定義
処理実行部と、
を備えたことを特徴とするファイルシステム。(1)

【 0 0 8 5 】

(付記 2)

付記 1 記載のファイルシステムに於いて、
前記ファイルをデータ領域とメタデータ領域との 2 つの領域で管理するファイル
管理部を有し、更に前記メタデータ領域には拡張メタデータ領域が設けられ、前
記ファイル管理部は、ユーザにより前記拡張メタデータ領域の形式を指定可能と
することを特徴とするファイルシステム。(2)

【 0 0 8 6 】

(付記 3)

付記 2 記載のファイルシステムに於いて、前記ファイル管理部は、前記データ
領域の内容に応じて前記拡張メタデータ領域の形式を指定することを特徴とする
ファイルシステム。(3)

【 0 0 8 7 】

(付記 4)

付記3記載のファイルシステムに於いて、前記ファイル管理部は、前記拡張メタデータ領域の形式としてファイルタイプというメタデータを設定し、前記ファイルタイプに応じて前記拡張メタデータ領域の形式を決定することを特徴とするファイルシステム。(4)

【0088】

(付記5)

付記4記載のファイルシステムに於いて、前記ファイル管理部は、ファイルの作成時にファイルタイプを決定すると共に該ファイルタイプに応じて拡張メタデータ領域を設定し、その後、ユーザによる前記ファイルタイプの変更と前記拡張メタデータ領域の変更を可能とすることを特徴とするファイルシステム。(5)

【0089】

(付記6)

付記5記載のファイルシステムに於いて、前記ファイル管理部は、前記ファイルのデータ領域の内容からファイルタイプを自動的に決定すると共に前記拡張メタデータ領域も自動的に決定することを特徴とするファイルシステム。

【0090】

(付記7)

付記5記載のファイルシステムに於いて、前記ファイル管理部は、複数ファイルを管理するためのディレクトリを用いたツリー構造を備え、ファイル作成時に設定されるファイルタイプは、親ディレクトリのファイルタイプを継承することを特徴とするファイルシステム。

【0091】

(付記8)

付記2記載のファイルシステムに於いて、前記ファイル管理部は、前記拡張メタデータ領域に、前記データ領域から抽出した情報を拡張メタデータとして保持することを特徴とするファイルシステム。

【0092】

(付記9)

付記8記載のファイルシステムに於いて、前記データ領域から抽出された前記

拡張メタデータは、前記データ領域と前記拡張メタデータ領域とに二重化して保持することを特徴とするファイルシステム。

【0093】

(付記10)

付記8記載のファイルシステムに於いて、前記データ領域から抽出した前記拡張メタデータは、実体は前記データ領域に持ち、前記拡張メタデータ領域には前記データ領域の実態へのポインタを保持していることを特徴とするファイルシステム。

【0094】

(付記11)

付記2記載のファイルシステムに於いて、前記定義処理実行部は、前記ファイルのデータ領域への書き込みの際に、前記ユーザ定義処理に基づき前記メタデータ領域の拡張メタデータを変更することを特徴とするファイルシステム。(6)

【0095】

(付記12)

付記2記載のファイルシステムに於いて、前記定義処理実行部は、前記ファイルのデータ領域への書き込みの際に、別途設けられたユーザプログラムにメッセージを送付して前記拡張メタデータ領域の拡張メタデータを変更させることを特徴とするファイルシステム。(7)

【0096】

(付記13)

付記2記載のファイルシステムに於いて、前記定義処理実行部は、前記ファイルのデータ領域への書き込みの際にデータ領域変更フラグを立て、別途設けられたユーザプログラムにより前記フラグが立ったことを契機として前記拡張メタデータ領域の拡張メタデータを変更させることを特徴とするファイルシステム。(8)

【0097】

(付記14)

付記4記載のファイルシステムに於いて、前記定義処理実行部は、前記ファイ

ルタイプに応じてユーザ定義処理を実行することを特徴とするファイルシステム

【 0 0 9 8 】

(付記 1 5)

付記 1 記載のファイルシステムに於いて、ユーザが処理を定義するための A P I を備えたことを特徴とするファイルシステム。

【 0 0 9 9 】

(付記 1 6)

付記 1 5 記載のファイルシステムに於いて、前記 A P I は、カーネル領域で実行される A P I とユーザ領域で実行される A P I との 2 層構造を備えたことを特徴とするファイルシステム。

【 0 1 0 0 】

(付記 1 7)

付記 2 記載のファイルシステムに於いて、前記拡張メタデータ領域のサイズはファイル内容に応じて可変であることを特徴とするファイルシステム。

【 0 1 0 1 】

(付記 1 8)

オペレーティングシステム上で動作するファイルシステムの制御方法に於いて

ユーザが予め定義したユーザ定義処理を保持する定義処理保持ステップと、
ファイルに対するアクセスが発生した場合に、該アクセスに応じて前記ファイル
を処理するアクセス実行ステップと、
前記ファイルへのアクセスを契機として、前記ユーザ定義処理を実行させる定義
処理実行ステップと、
を有することを特徴とするファイルシステムの制御方法。(9)

【 0 1 0 2 】

(付記 1 9)

付記 1 8 記載のファイルシステムの制御方法に於いて、
前記ファイルをデータ領域とメタデータ領域との 2 つの領域で管理し、更に前記

メタデータ領域には拡張メタデータ領域が設けられ、ユーザにより前記拡張メタデータ領域の形式を指定可能とするファイル管理ステップを備えたことを特徴とするファイルシステムの制御方法。

【 0 1 0 3 】

(付記 2 0)

付記 1 9 記載のファイルシステムの制御方法に於いて、前記ファイル管理ステップは、前記データ領域の内容に応じて前記拡張メタデータ領域の形式を指定することを特徴とするファイルシステムの制御方法。

【 0 1 0 4 】

(付記 2 1)

付記 2 0 記載のファイルシステムの制御方法に於いて、前記ファイル管理ステップは、前記拡張メタデータ領域の形式としてファイルタイプというメタデータを設定し、前記ファイルタイプに応じて前記拡張メタデータ領域の形式を決定することを特徴とするファイルシステムの制御方法。

【 0 1 0 5 】

(付記 2 2)

付記 2 1 記載のファイルシステムの制御方法に於いて、前記ファイル管理ステップは、ファイルの作成時にファイルタイプを決定すると共に該ファイルタイプに応じた拡張メタデータ領域を設定し、その後、ユーザによる前記ファイルタイプの変更と前記拡張メタデータ領域の変更を可能とすることを特徴とするファイルシステムの制御方法。

【 0 1 0 6 】

(付記 2 3)

付記 2 2 記載のファイルシステムの制御方法に於いて、前記ファイル管理ステップは、前記ファイルのデータ領域の内容からファイルタイプを自動的に決定すると共に前記拡張メタデータ領域も自動的に決定することを特徴とするファイルシステムの制御方法。

【 0 1 0 7 】

(付記 2 4)

付記 2 2 記載のファイルシステムの制御方法に於いて、前記ファイル管理ステップは、複数ファイルを管理するためのディレクトリを用いたツリー構造を備え、ファイル作成時に設定されるファイルタイプは、親ディレクトリのファイルタイプを継承することを特徴とするファイルシステムの制御方法。

【 0 1 0 8 】

(付記 2 5)

付記 1 9 記載のファイルシステムの制御方法に於いて、前記ファイル管理ステップは、前記拡張メタデータ領域に、前記データ領域から抽出した情報を拡張メタデータとして保持することを特徴とするファイルシステムの制御方法。

【 0 1 0 9 】

(付記 2 6)

付記 2 5 記載のファイルシステムの制御方法に於いて、前記データ領域から抽出された前記拡張メタデータは、前記データ領域と前記拡張メタデータ領域とに二重化して保持することを特徴とするファイルシステムの制御方法。

【 0 1 1 0 】

(付記 2 7)

付記 2 5 記載のファイルシステムの制御方法に於いて、前記データ領域から抽出した前記拡張メタデータは、実体は前記データ領域に持ち、前記拡張メタデータ領域には前記データ領域の実態へのポインタを保持していることを特徴とするファイルシステムの制御方法。

【 0 1 1 1 】

(付記 2 8)

付記 1 9 記載のファイルシステムの制御方法に於いて、前記定義処理実行ステップは、前記ファイルのデータ領域への書き込みの際に、前記ユーザ定義処理に基づき前記メタデータ領域の拡張メタデータを変更することを特徴とするファイルシステムの制御方法。

【 0 1 1 2 】

(付記 2 9)

付記 1 9 記載のファイルシステムの制御方法に於いて、前記定義処理実行ステ

ップは、前記ファイルのデータ領域への書き込みの際に、別途設けられたユーザプログラムにメッセージを送付して前記拡張メタデータ領域の拡張メタデータを変更させることを特徴とするファイルシステムの制御方法。

【0113】

(付記30)

付記19記載のファイルシステムの制御方法に於いて、前記定義処理実行ステップは、前記ファイルのデータ領域への書き込みの際にデータ領域変更フラグを立て、別途設けられたユーザプログラムにより前記フラグが立ったことを契機として前記拡張メタデータ領域の拡張メタデータを変更させることを特徴とするファイルシステムの制御方法。

【0114】

(付記31)

付記21記載のファイルシステムの制御方法に於いて、前記定義処理実行ステップは、前記ファイルタイプに応じてユーザ定義処理を実行することを特徴とするファイルシステムの制御方法。

【0115】

(付記32)

付記19記載のファイルシステムの制御方法に於いて、ユーザが処理を定義するためのAPIを備えたことを特徴とするファイルシステムの制御方法。

【0116】

(付記33)

付記32記載のファイルシステムの制御方法に於いて、前記APIは、カーネル領域で実行されるAPIとユーザ領域で実行されるAPIとの2層構造を備えたことを特徴とするファイルシステムの制御方法。

【0117】

(付記34)

付記19記載のファイルシステムの制御方法に於いて、前記拡張メタデータ領域のサイズはファイル内容に応じて可変であることを特徴とするファイルシステムの制御方法。

【0118】

(付記35)

コンピュータに、

ユーザが予め定義したユーザ定義処理を保持する定義処理保持ステップと、

ファイルに対するアクセスが発生した場合に、該アクセスに応じて前記ファイルを処理するアクセス実行ステップと

前記ファイルへのアクセスを契機として、前記ユーザ定義処理を実行させる定義処理実行ステップと、

を実行させることを特徴とするプログラム。(10)

【0119】

【発明の効果】

以上説明してきたように本発明によれば、ファイルに対するアクセスを契機にユーザが定義した処理をファイルシステムが直接実行することとなり、アプリケーションに依存することなく、ファイルへのアクセスを契機にユーザが定義した処理が実行でき、より柔軟なデータ制御を可能とし、またファイルシステムの性能向上を図ることができる。

【図面の簡単な説明】

【図1】

本発明の原理説明図

【図2】

本発明によるファイルシステムの機能構成のブロック図

【図3】

本発明のファイルシステムとハードウェアの関係の説明図

【図4】

ファイルアクセスに対する本発明の処理動作の説明図

【図5】

本発明によるユーザ定義処理をアプリケーションを使用する従来例と対比した説

明図

【図 6】

本発明によるファイルシステムの処理手順のフローチャート

【図 7】

ユーザの定義により拡張メタデータ領域の形式を変更する本発明の実施形態の説明図

【図 8】

拡張メタデータ領域のファイルタイプをユーザの定義により変更する実施形態の説明図

【図 9】

図 8 のファイル管理に用いるファイルタイプ構造体の説明図

【図 1 0】

図 8 のファイルタイプ構造体に従った追加削除によるリスト処理の明図

【図 1 1】

拡張メタデータを二重化して保持する実施形態の説明図

【図 1 2】

拡張メタデータをポインタで保持する実施形態の説明図

【図 1 3】

ユーザ定義処理の実行により直接に拡張メタデータを処理する実施形態の説明図

【図 1 4】

ユーザ定義処理から外部のプログラムにメッセージを伝えて拡張メタデータを処理する実施形態の説明図

【図 1 5】

ユーザ定義処理によりフラグを立てて外部のプログラムにより拡張メタデータを処理する実施形態の説明図

【図 1 6】

拡張メタデータ領域を拡大する処理の説明図

【図 1 7】

本発明のユーザ定義処理の設定に使用する 2 層構造をもつ A P I の説明図

【図 1 8】

メールサーバでメールを受けた際にユーザが定義した処理を実行する本発明の具体的な実施形態の説明図

【符号の説明】

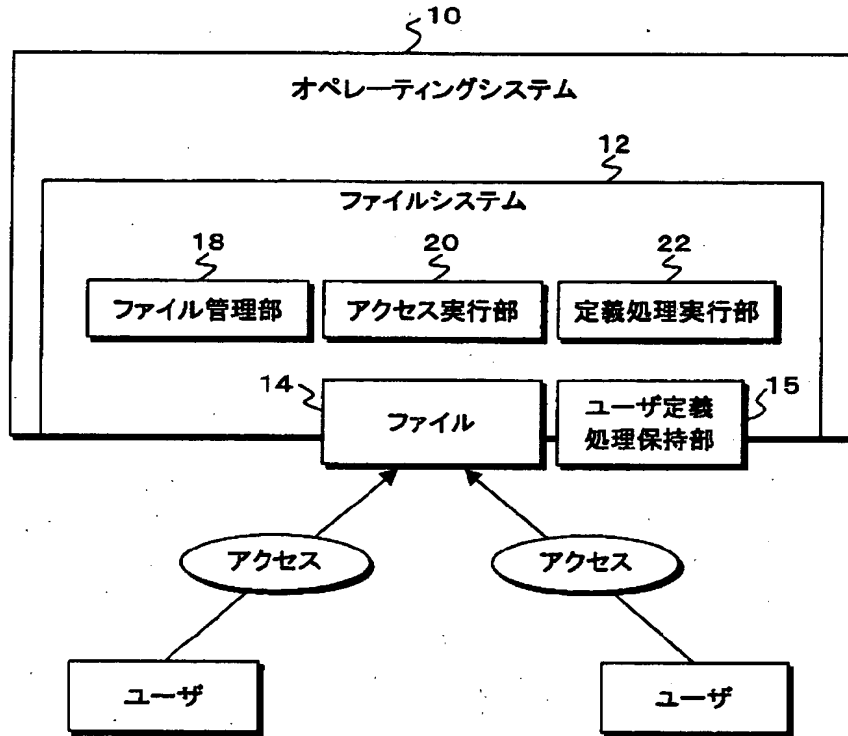
- 1 0 : オペレーティングシステム
- 1 2 : ファイルシステム
- 1 4 : ファイル
- 1 5 : ユーザ定義処理保持部
- 1 6, 1 6 - 1 ~ 1 6 - 3 : ユーザ定義処理
- 1 8 : ファイル管理部
- 2 0 : アクセス実行部
- 2 2 : 定義処理実行部
- 2 4 - 1, 2 4 - 2 : ユーザ
- 2 5 : 共通 A P I
- 2 6 - 1 ~ 2 6 - n : ドライバ
- 2 8 - 1 ~ 2 8 - n : ハードウェア
- 3 0 : データ領域
- 3 2 : メタデータ領域
- 3 4 : 拡張メタデータ領域
- 3 6 - 1 ~ 3 6 - 3 : 拡張メタデータ
- 7 0, 8 6 : ユーザプログラム

【書類名】 図面

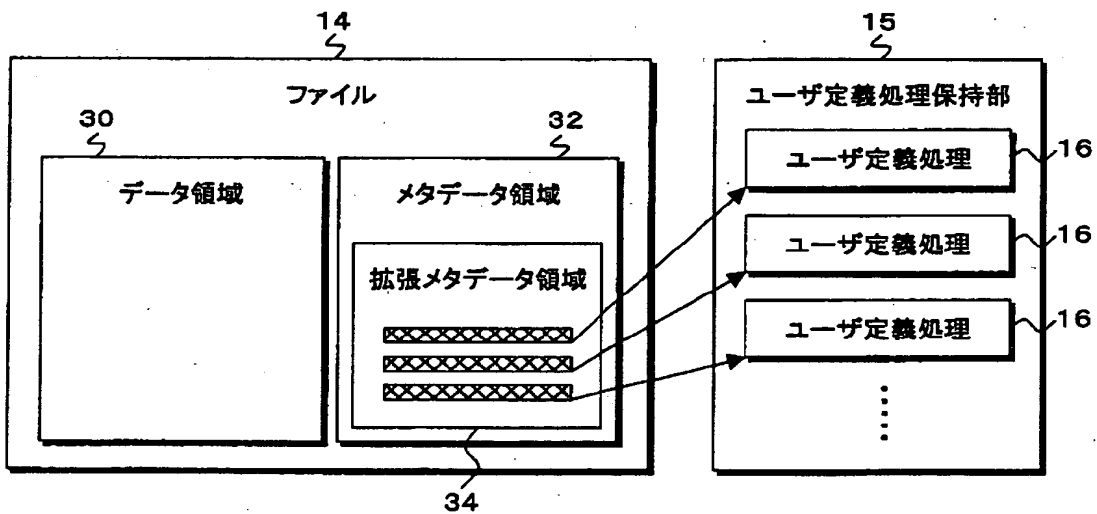
【図 1】

本発明の原理説明図

(A)

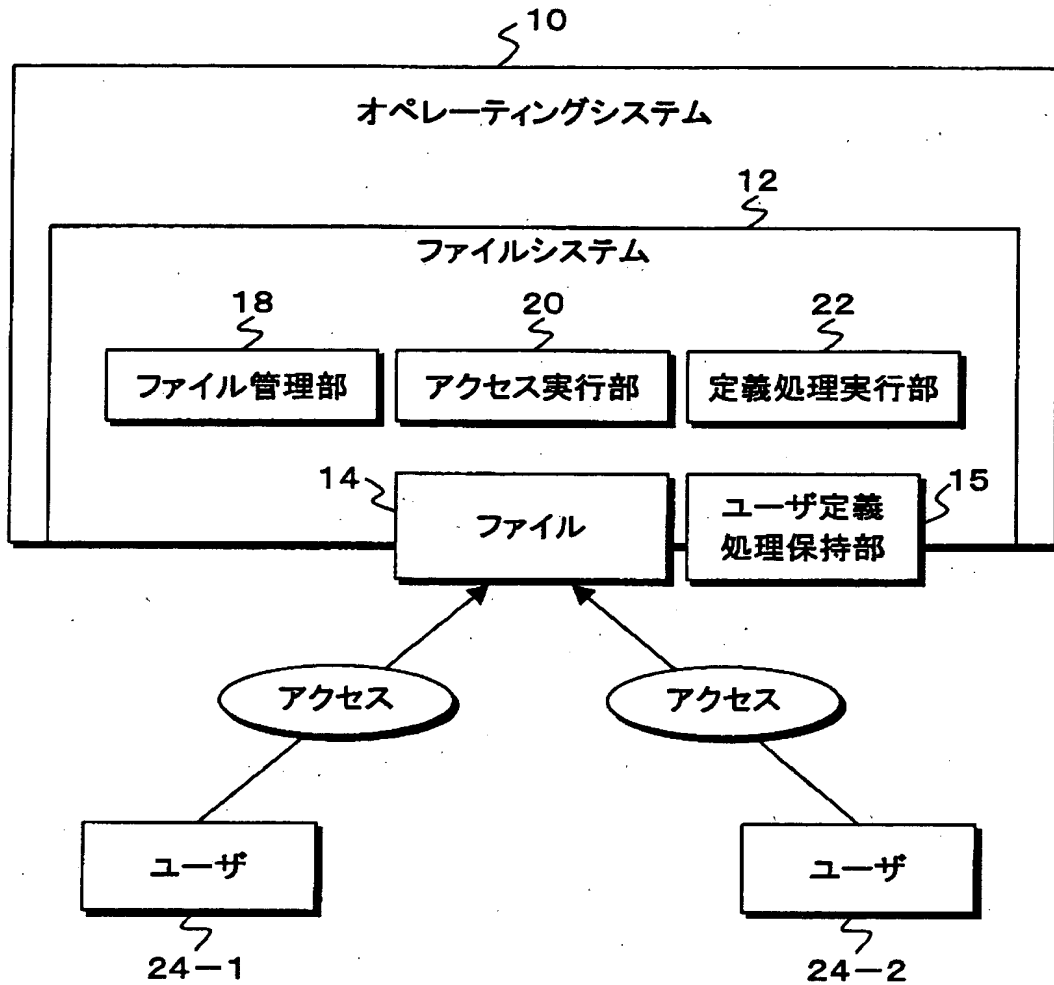


(B)



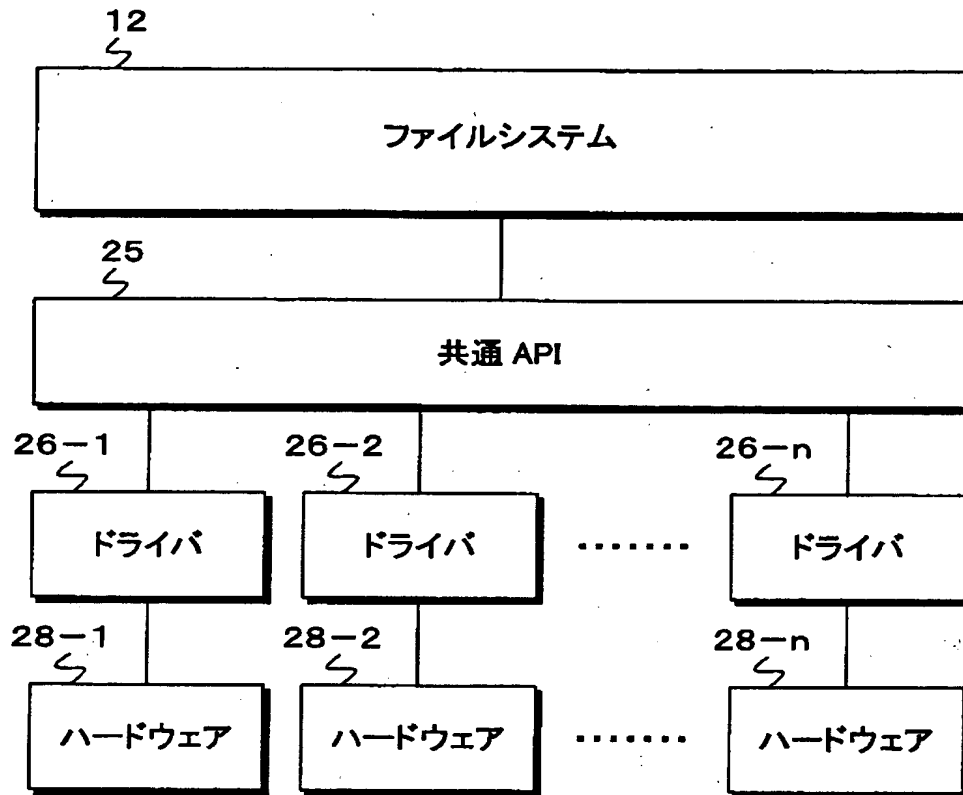
【図 2】

本発明によるファイルシステムの機能構成のブロック図



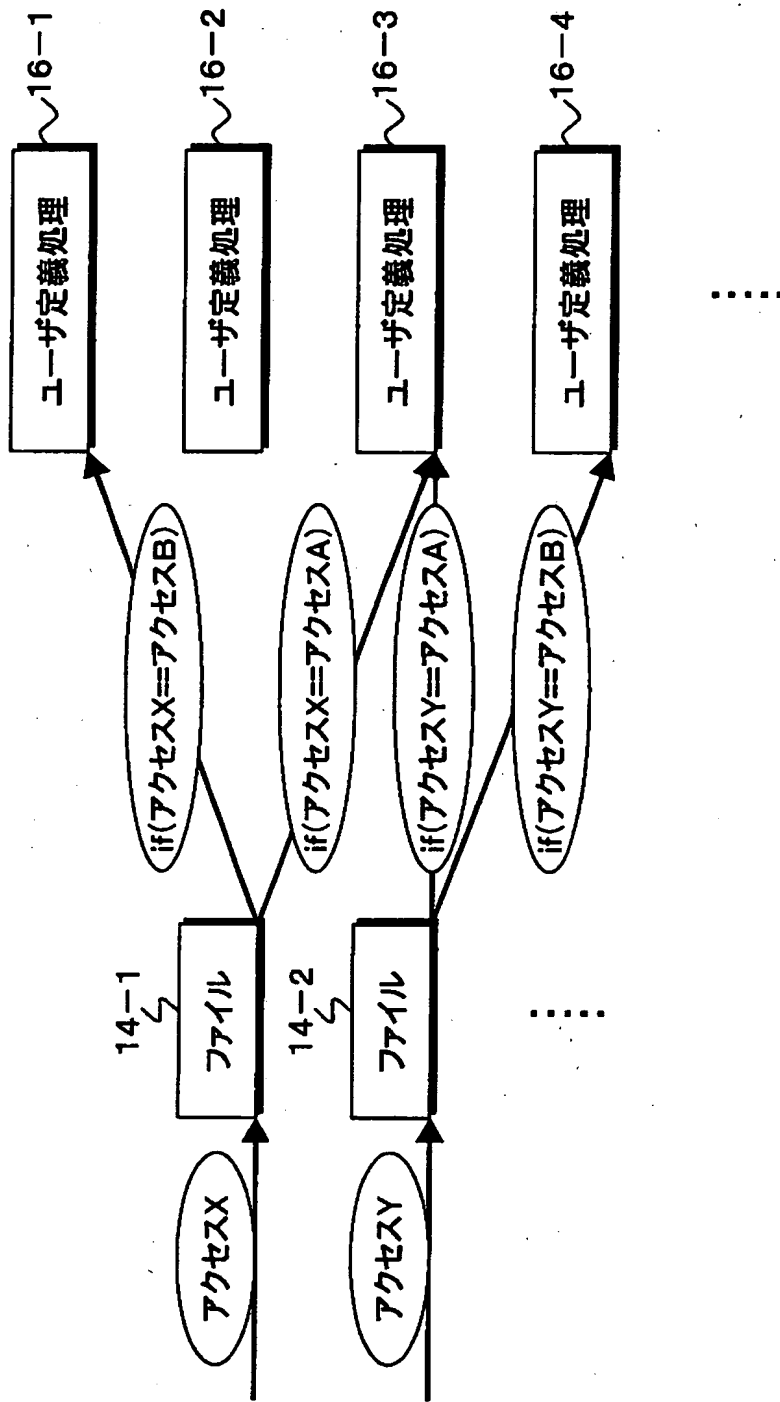
【図 3】

本発明のファイルシステムとハードウェアの関係の説明図



【図 4】

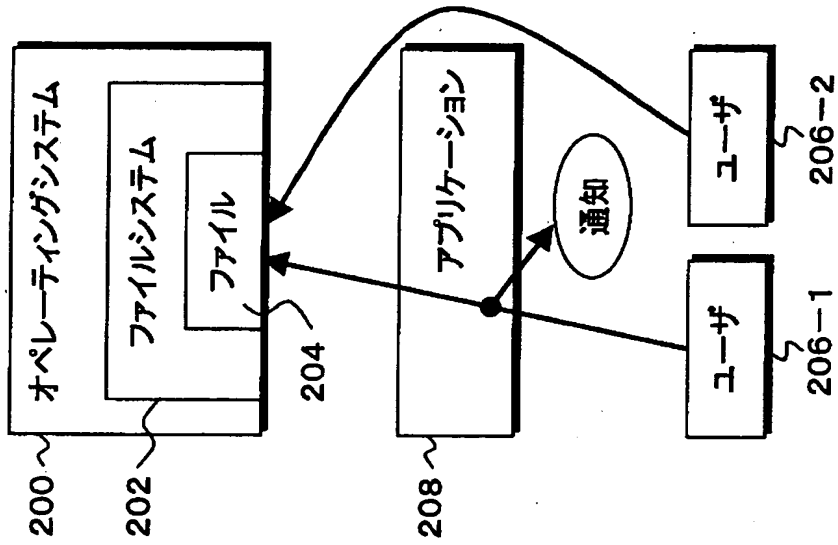
ファイルアクセスに対する本発明の処理動作の説明図



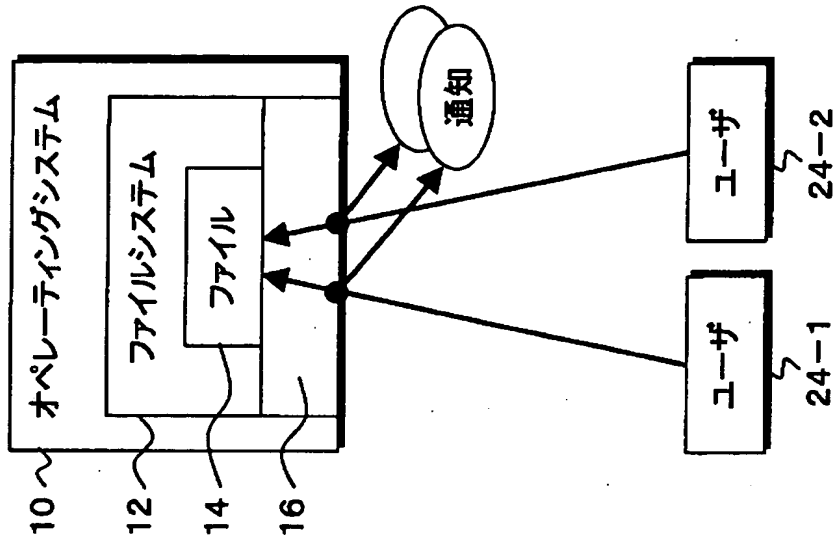
【図 5】

本発明によるユーザ定義処理をアプリケーションを使用する従来例と対比した説明図

(A)

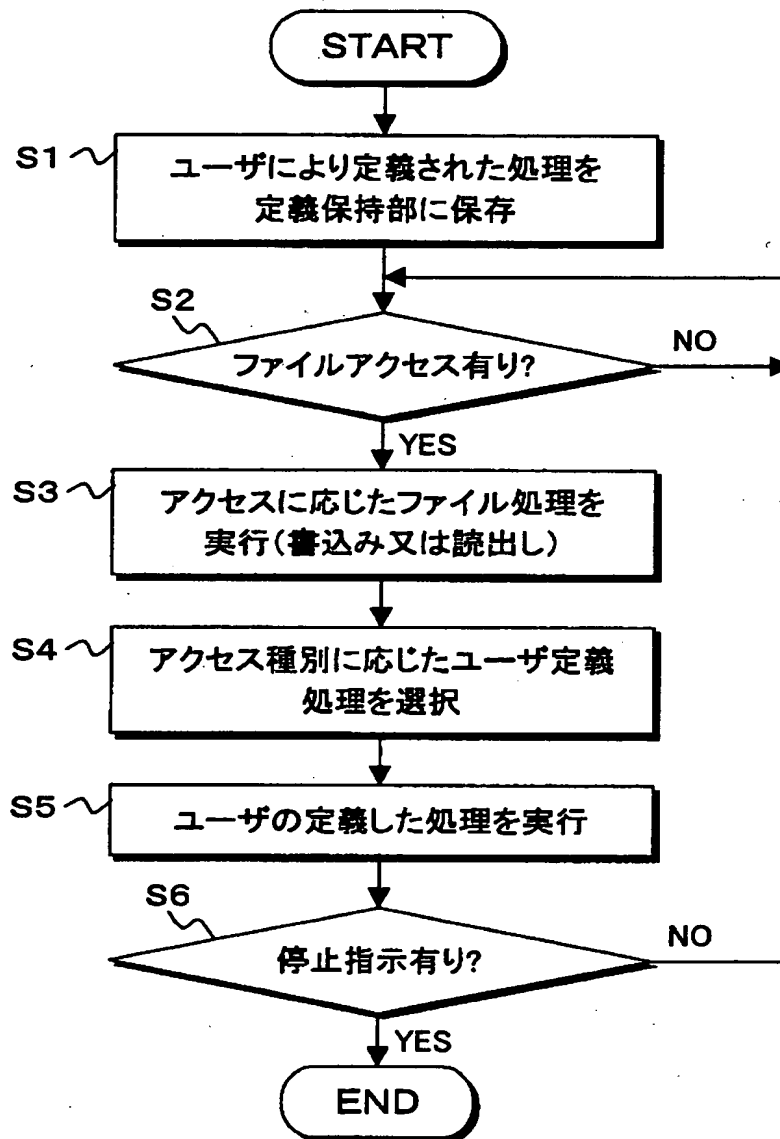


(B)



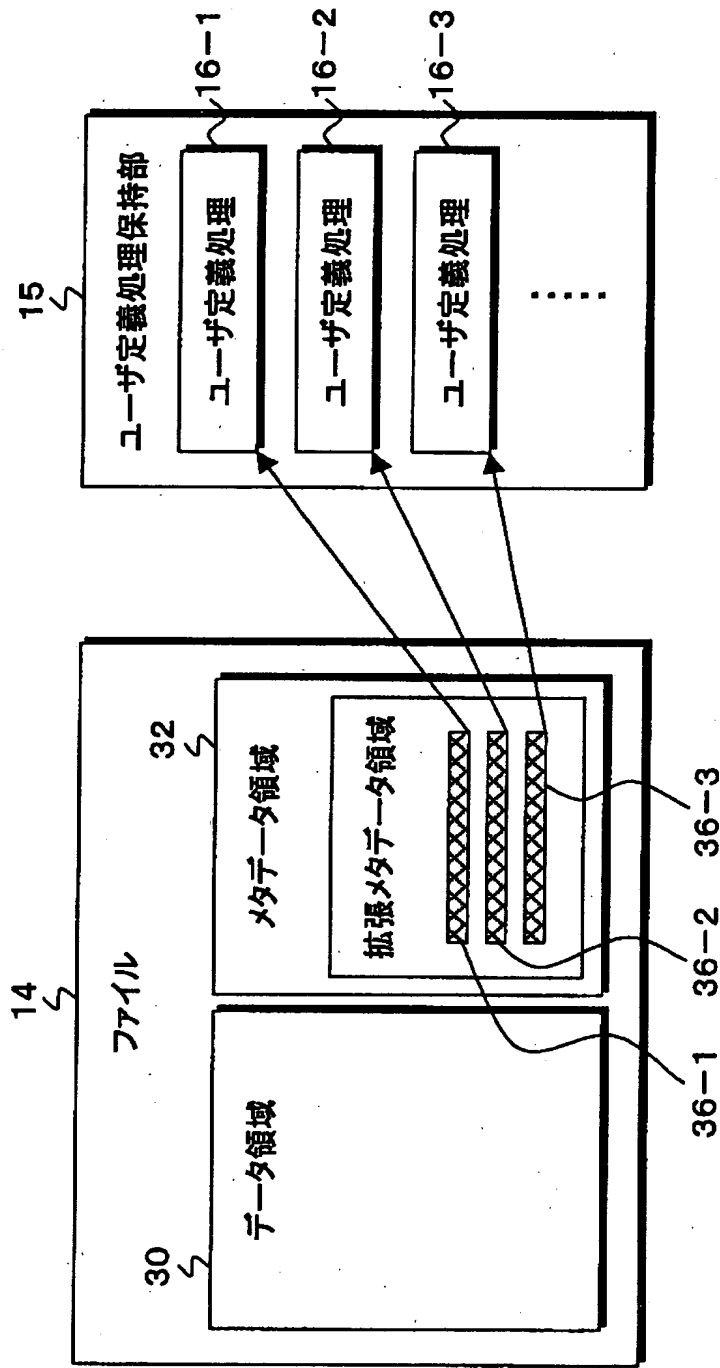
【図 6】

本発明によるファイルシステムの処理手順のフローチャート



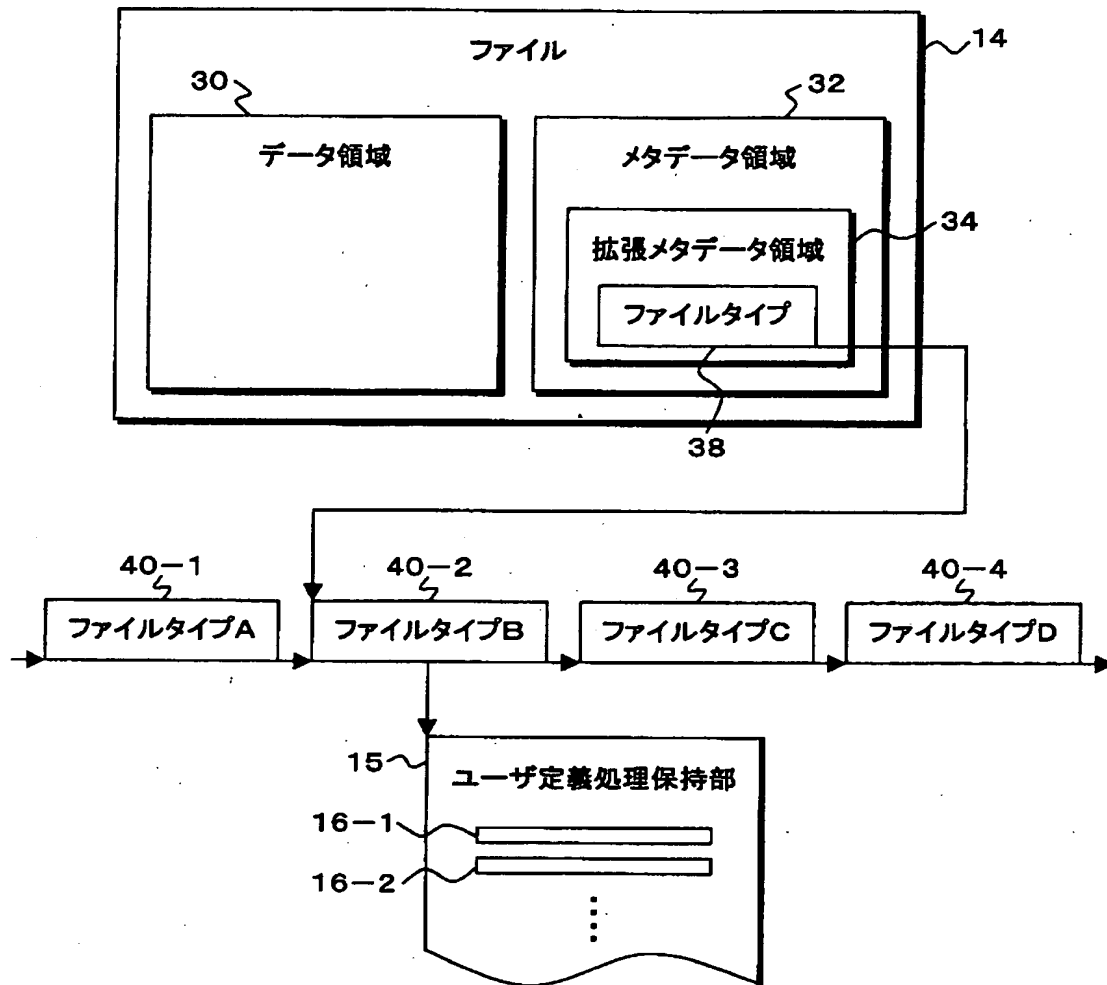
【図 7】

ユーザの定義により拡張メタデータ領域の形式を変更する本発明の実施形態の説明図



【図 8】

拡張メタデータ領域のファイルタイプをユーザの定義により変更する実施形態の説明図



【図9】

図8のファイル管理に用いるファイルタイプ構造体の説明図

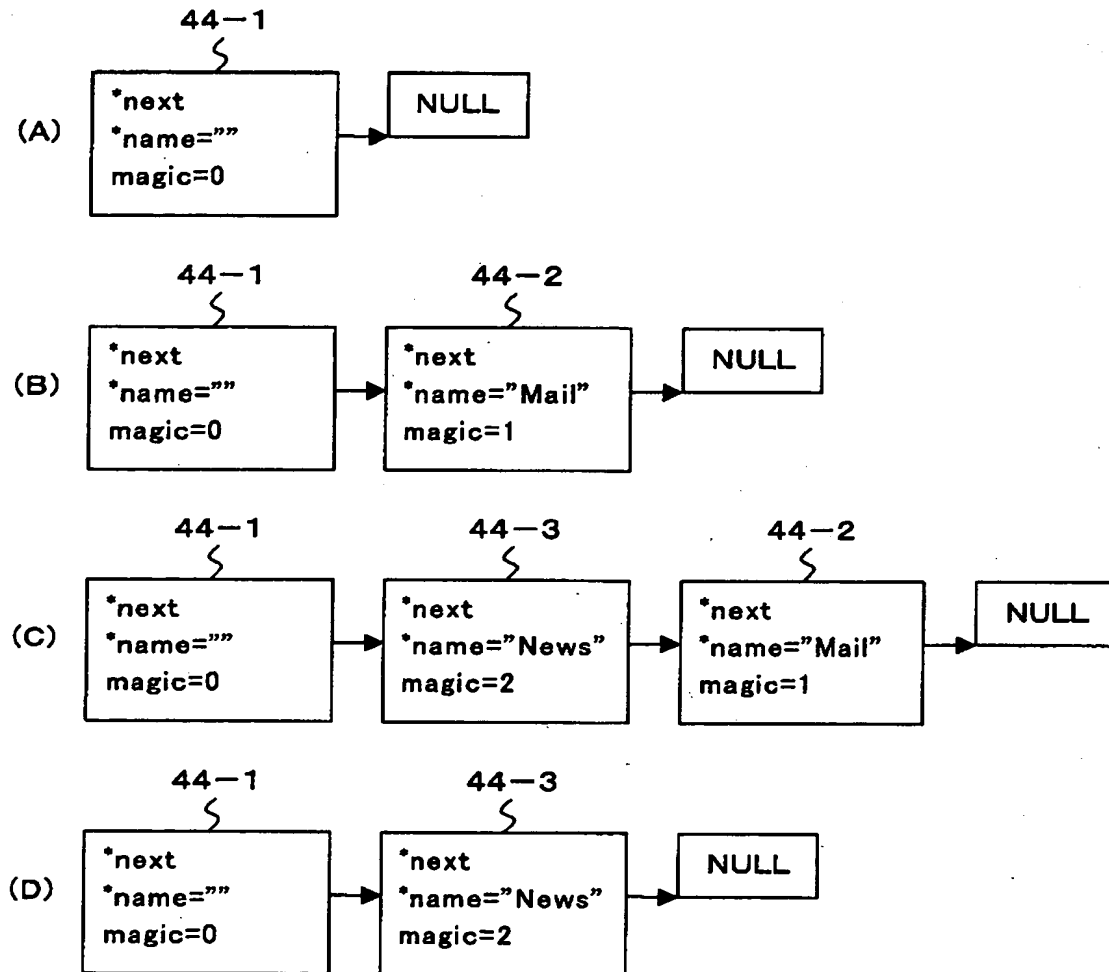
42

```

struct file_type_name {
    struct file_type_name *next;
    char *name;
    int magic;
};
    
```

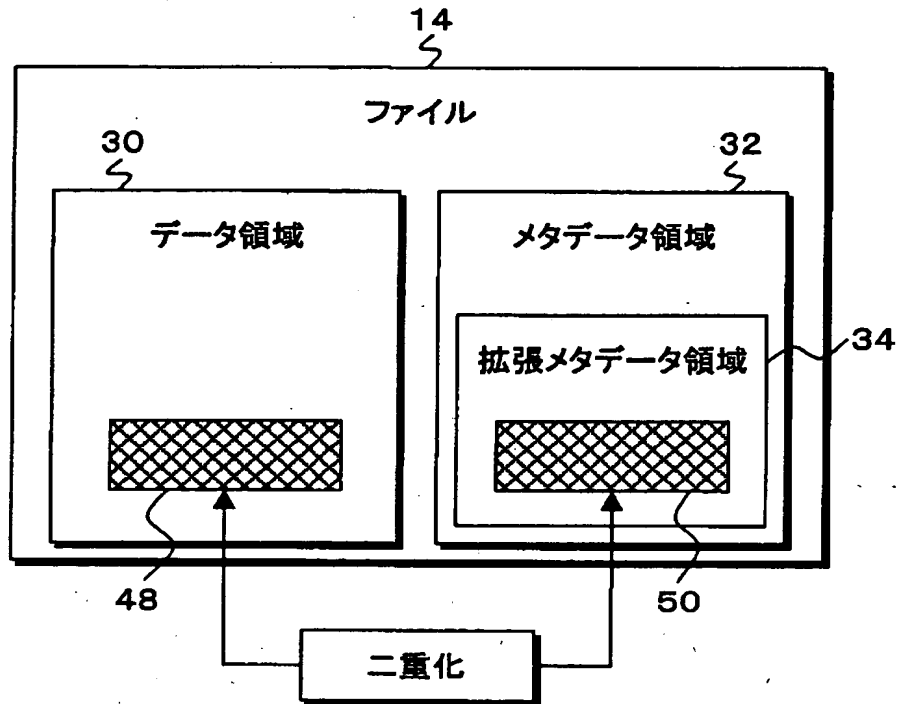
【図 10】

図8のファイルタイプ構造体に従った追加削除によるリスト処理の説明図



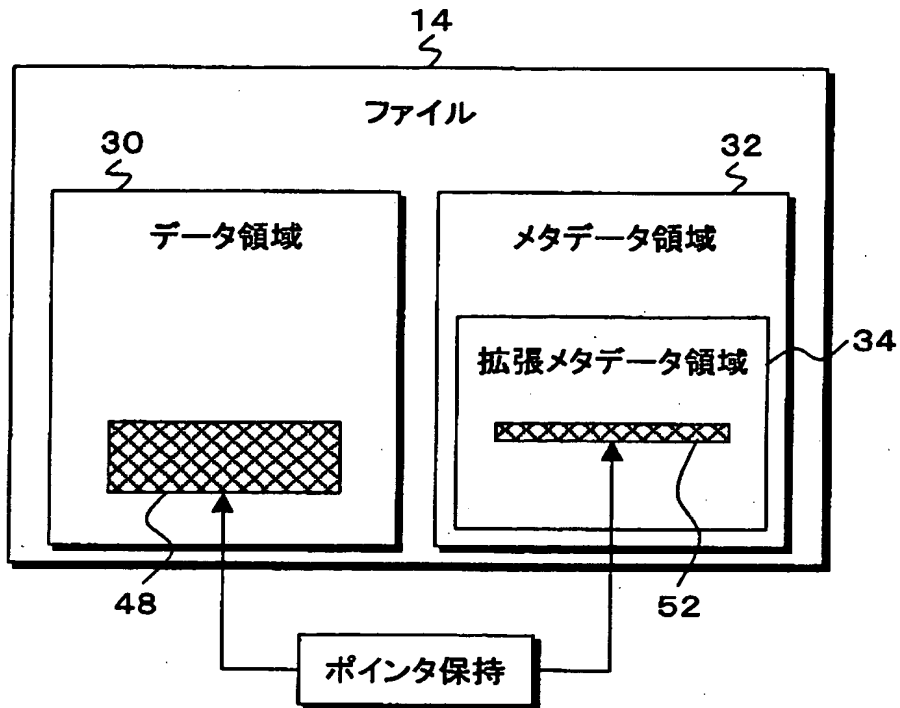
【図 1 1】

拡張メタデータを二重化して保持する実施形態の説明図



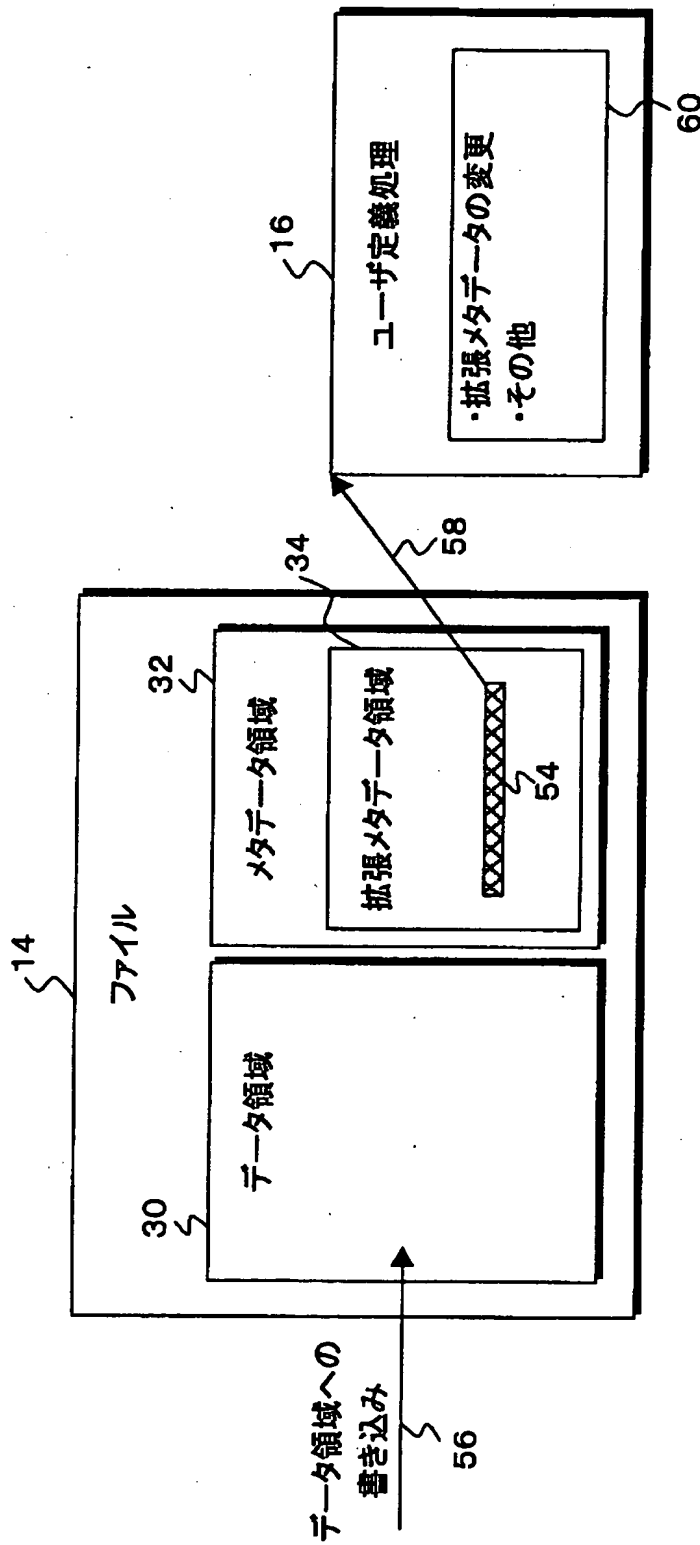
【図 1 2】

拡張メタデータをポインタで保持する実施形態の説明図



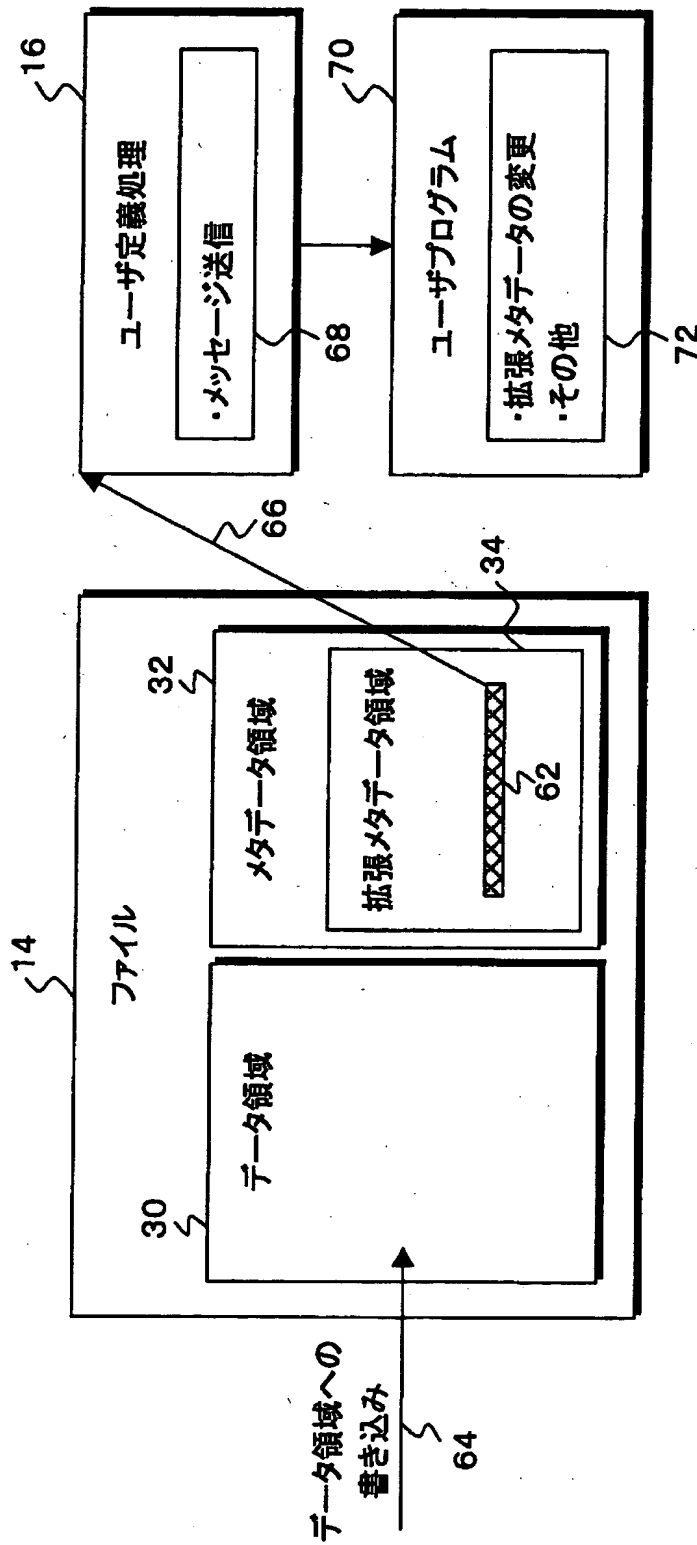
【図 13】

ユーザ定義処理の実行により直接に拡張メタデータを処理する実施形態の説明図



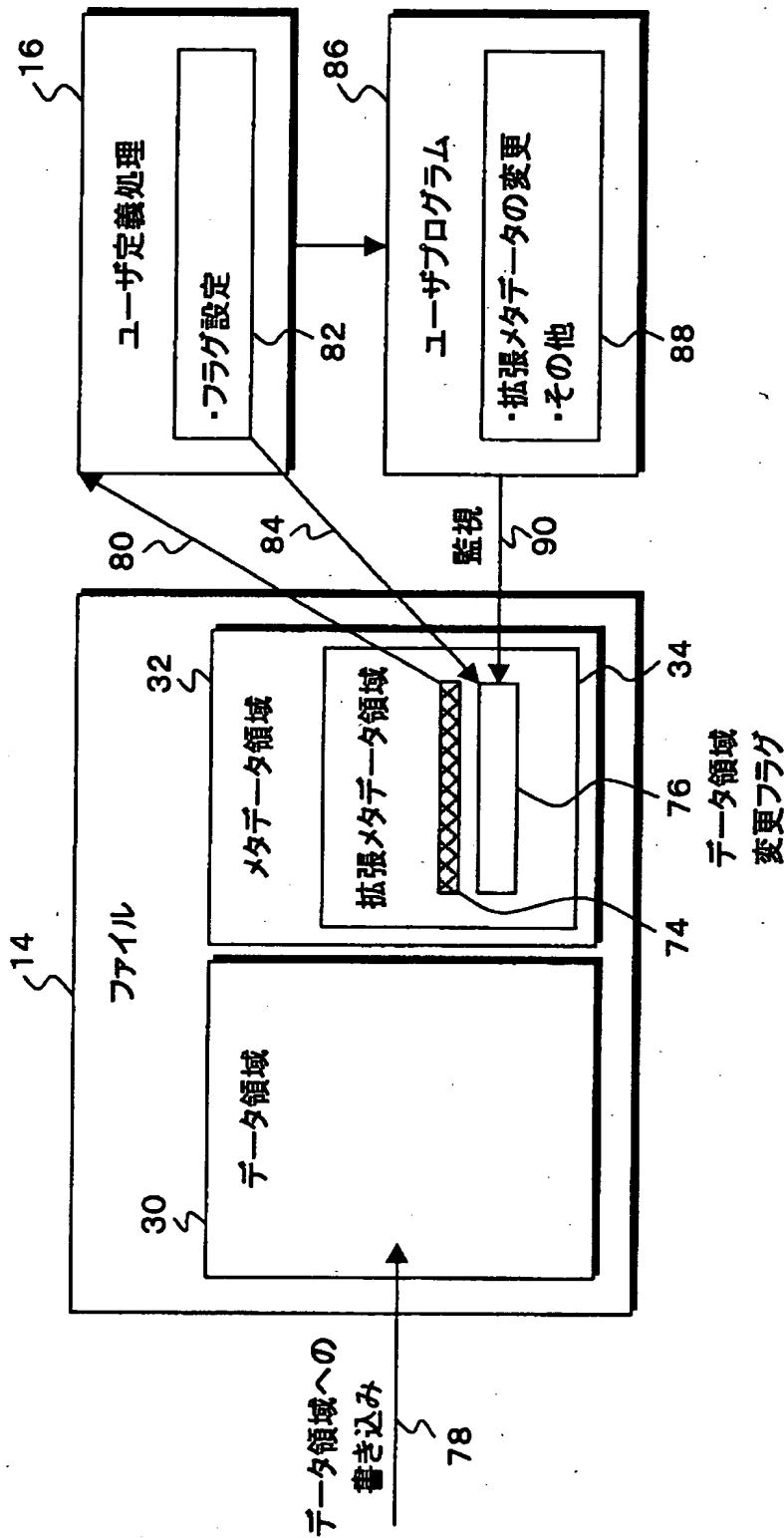
【図 14】

ユーザ定義処理から外部のプログラムにメッセージを伝えて拡張メタデータを処理する実施形態の説明図

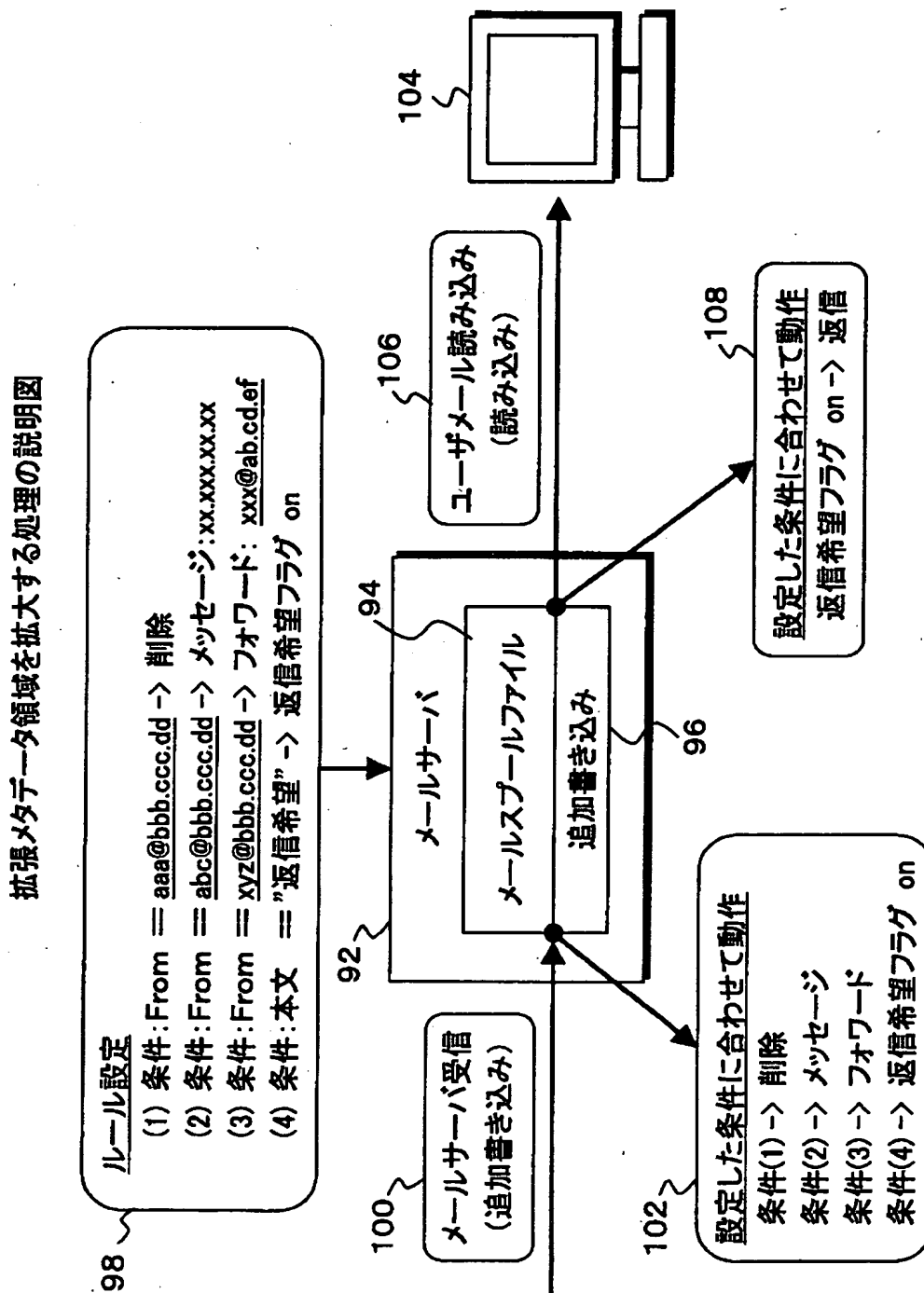


【図 15】

ユーザ定義処理によりフラグを立てて外部のプログラムにより拡張メタデータを処理する実施形態の説明図

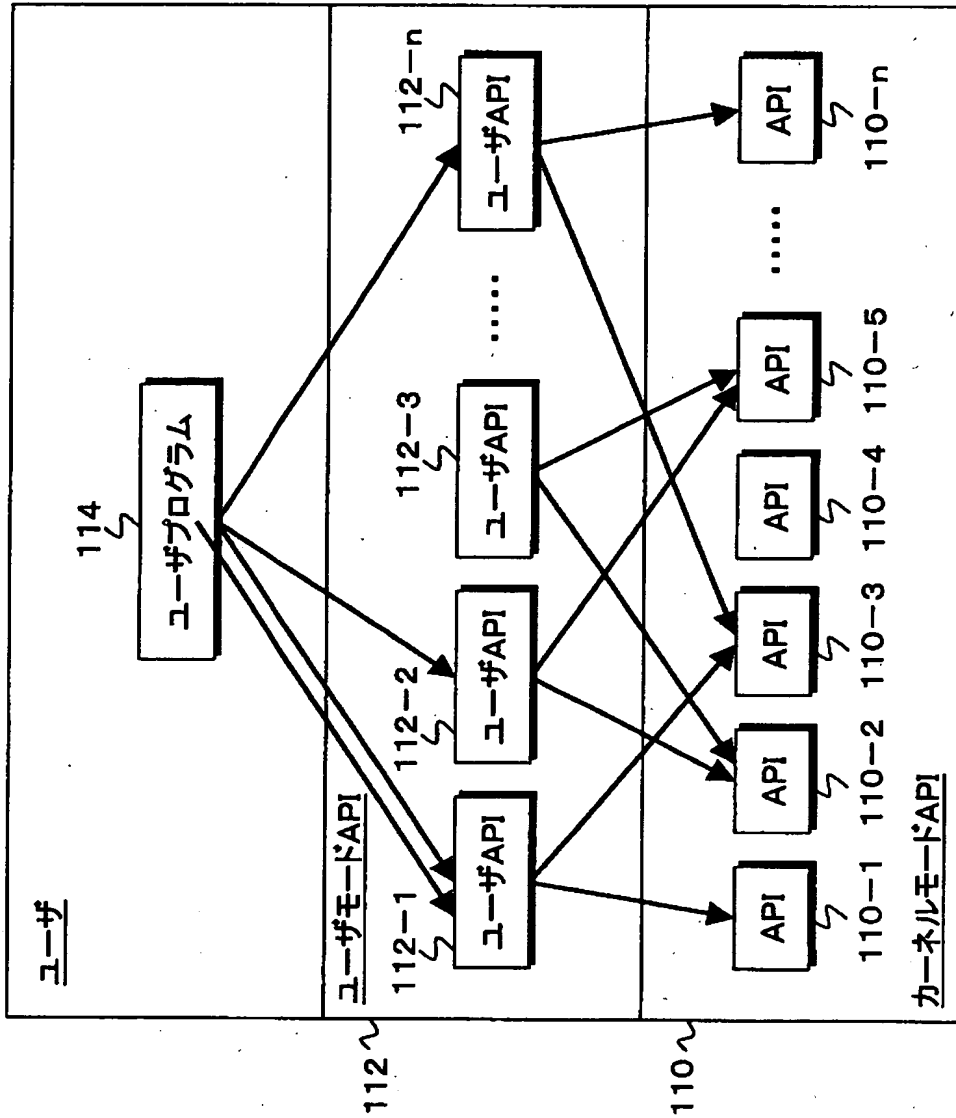


【図 16】



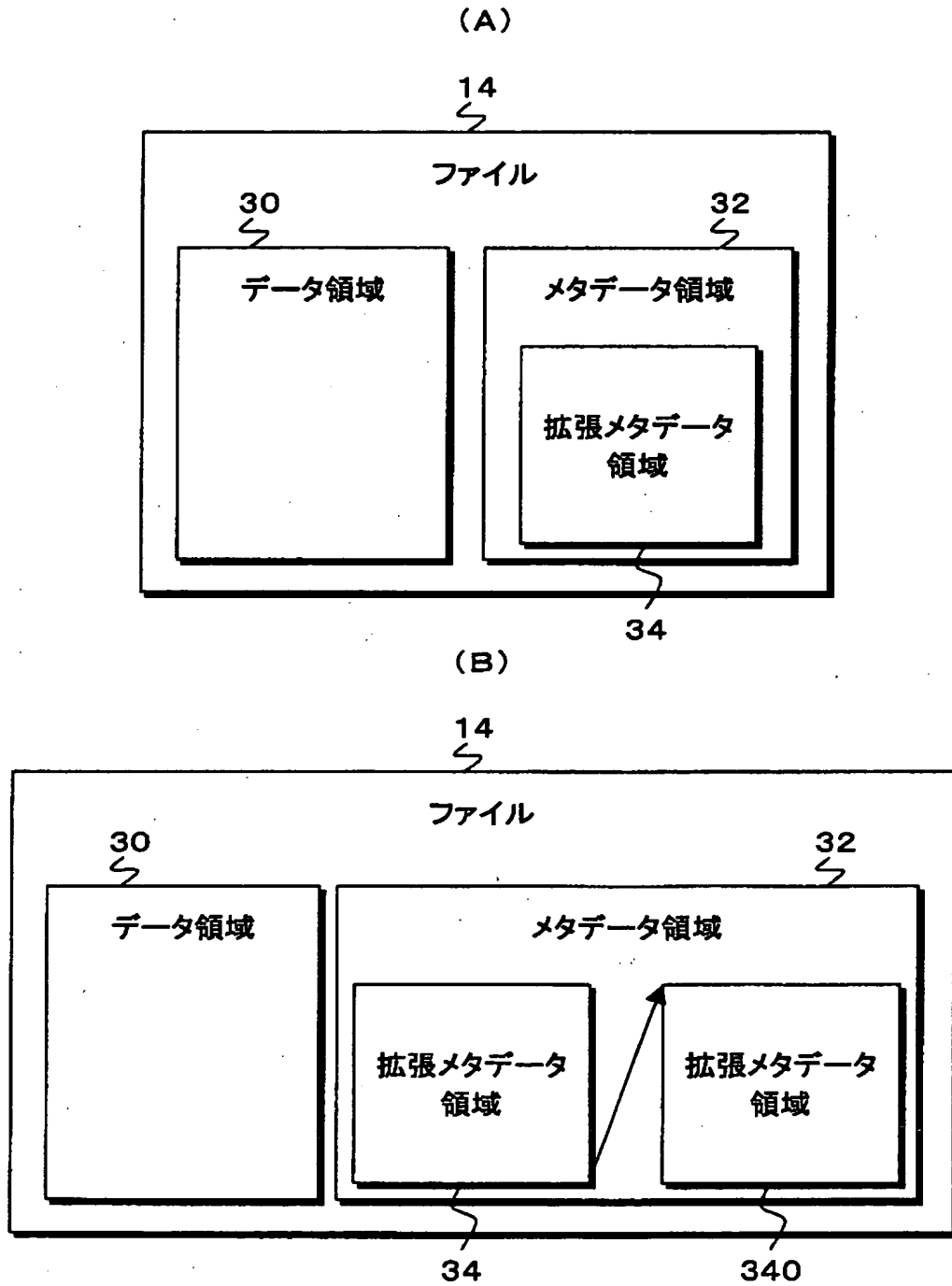
【図 17】

本発明のユーザ定義処理の設定に使用する2層構造をもつAPIの説明図



【図 18】

メールサーバでメールを受けた際にユーザが定義した処理を実行する
本発明の具体的な実施形態の説明図



【書類名】 要約書

【要約】

【課題】 アプリケーションに依存することなくファイルへのアクセスを契機にユーザが定義した処理を可能とする。

【解決手段】 オペレーティングシステム上で動作するファイルシステム12は、ファイル14に対する書込み又は読出しといったアクセスが発生した場合に、アクセスに応じてファイル14を処理するアクセス実行部20と、ユーザが予め定義したユーザ定義処理を保持するユーザ理保持部15、ファイル14へのアクセスを契機としてユーザ定義処理を実行させる定義処理実行部22を備える。

【選択図】 図1

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日
[変更理由] 住所変更
住 所 神奈川県川崎市中原区上小田中4丁目1番1号
氏 名 富士通株式会社